

Exemple de départ – intégrale approchée d’une fonction

Le but est de calculer l’intégrale approchée d’une fonction dans l’intervalle $[0,1]$.

L’algorithme est la méthode des trapèzes sur l’intervalle divisé en **n** sous-intervalles.

La fonction à intégrer est passée implicitement à la fonction qui calcule (version 1a) et comme pointeur de fonction (version 1b).

Pour compiler:

Taper:

```
./build.py
```

La compilation crée 2 exécutables.

Pour exécuter:

```
./install/execbase1a --n N --x0 X0 -- x1 X1  
./install/execbase1b --n N --x0 X0 -- x1 X1
```

où les options sont:

- **N** est le nombre de sous-intervalles, par défaut 100000
- **X0** est le début de l’intervalle d’intégration, par défaut 0.0
- **X1** est la fin de l’intervalle d’intégration, par défaut 1.0
- calcule l’intégrale approchée entre **X0** et **X1**.